Numbers

Objective

This projectlet will use recreational number theory as a means to understand performance considerations. As the “problem size” grows the solution cost - time and/or resources grow as well but a judicious choice of algorithms and data structures will determine the viability of the solution.

In addition, this projectlet will encourage a layered approach to the solution. A core layer with a clean interface specification will support other higher layers with an eventual top layer to solve the specific problem.

REFERENCE

There will be many terms used in this projectlet that would require precise definitions. Following sites could be used as references.

https://www.wikipedia.org/
https://www.wolframalpha.com/

User needs and requirements

Library

Id	Need/Requirement
1	A library to support explorations in recreational number theory is needed with the following core routines.
2	Given a number a routine to return a list of decimal digits
3	Given a list of decimal digits, a routine to return the value
4	Given a number a routine to return a list of all divisors
5	Given a number a routine to return a list of its prime factors
6	Given a list of numbers, a routine to return the sum of all the numbers
7	Given a list of numbers, the product of all of the numbers
8	Given a list of numbers, to return a list of all squares, cubes, … other powers
9	Given 2 numbers, to return the greatest common divisor

Id	Need/Requirement
10	Generate a series of Fibonacci numbers
11	Given an array of numbers generate pairwise gcd's
12	Generate a series of taxicab numbers of a specified order

Applications

Id	Need/Requirement
1	Given a number - report if it is prime
2	Given a number - report if it is perfect
3	Given a number - report if it is a Kaprekar number
4	Given a number - report if it is a Harshad number
5	For a list of Fibonacci series, report the gcd of all pairs of numbers
6	Enumerate taxicab numbers (Ramanujan-Hardy numbers) of the order 3, 4

User wants

1	A density graph of primes (e.g. no in each range of 1000)
2	A density graph of fibonacci numbers

Example usage

SIMPLE TESTS ON NUMBERS

```
No 4096------------------------
DigitsOf
[4 0 9 6 ]
Value is 4096
DivisorsOf
[1 2 4 8 16 32 64 64 128 256 512 1024 2048 4096 ]
PrimeFactorsOf
[1 2 2 2 2 2 2 2 2 2 2 2 ]
Product of all those 4096
IsPrime False
IsPerfect False
IsHarshad False
IsHappy True
Iskaprekar False
```

PAIRWISE GCD OF FIBONACCI NUMBER SERIES

```
1043008345   527452805    : 5
1043008345   132996290    : 5045
1043008345   1640061041   : 29
1043008345   1511089245   : 5
1043008345   2001647585   : 5
1043008345   1479474520   : 5
1043008345   66908715     : 5
1043008345   1669358405   : 5
1043008345   602368435    : 5
1043008345   1528872610   : 5
1043008345   532750075    : 5
1043008345   318507655    : 5
1043008345   872122360    : 5
1043008345   196167680    : 5
1043008345   848533765    : 145
1043008345   2111895522   : 1009
1043008345   1675338205   : 5
1043008345   1739456245   : 5
1043008345   1597594320   : 5
```

TIMING COMPARISONS FOR PAIRWISE GCD COMPUTATION

1024 ELEMENTS

```
real 0m0.352s
user 0m0.149s
sys  0m0.060s
```

4096 ELEMENNTS

```
real.  0m4.146s
user0m1.754s
sys  0m0.562s
```

TAXICAB NUMBERS OF ORDER 3

```
7872536
        184      118
        196      70
7957504
        184      120
        198      58
8077888
        186      118
        192      100
```

TAXICAB NUMBERS OF ORDER 4

```
635318657
        134      133
        158      59
```